Report from Break-Out Discussion (2/2)

Models@run.time workshop, October 3, 2006, Genova

What should a run-time model look like?

- Normally the models of a system is represented as a state at run time. But for Nelly: from the point of view of the adaptive system how can we produce a model which will guide the model at transforming from one state to another, ensuring that the model is in never in the wrong state?
- Q: But what is the medatada that we need to include into such a model?
- A: The model will depend on the domain that we are modelling, so each run-time model will be defined in terms of the elements we need to model.
- S: develop a list of questions/questionnaire to guide in run-time model development, specialised per domain.

What should a run-time model look like?

- What is the defining features of the run-time model vs. design model?
  - Instance model at run-time, vs class model at design time;
- Run-time model is there for "reflection" , i.e. it is an ongoing representation of the system that is running, its execution. Including several instances of the design-level class, as well as and the state.
  - The model is updated by following the current execution and when there is a need to interrupt/correct the execution, we need to update/correct the model. In other words, the run-time model is continuously updated.
  - Depending on the purpose of the system modelled, we need to specialise the models for the that specific goal, e.g. for monitoring we need observation.
- If looking from the meta-modelling perspective, the run-time model is no different from any other model.

How can the models be maintained at run-time?

- The objectives of the model will also affect the way the models are maintained. E.g., for instance when modelling adaptation, we need to have some adaptation operators. But the level of the operator will be defined by the type of adaptation required.

- What is *Maintain*:
  – Keep up the correct casually connected representation (synchronisation of the execution and representation)
  – How to establish direct mapping between elements

- Model is a representation of the system, then we need a meta-model to update the running of this model. (Fabio to explain).

Why to have models at run-time?

⇩
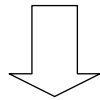
- The models are already there, i.e. there are already "abstractions", maybe not always very well understood and crude
- Controllers need to understand and control the system, so need to have a way to "abstract" for understanding and to interact with the system

⇧
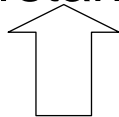
What is their role in Validation?

What are the best overall model-driven approaches for adaptive and autonomous systems?

- Need a sufficient set of concerns (in other words also a technique for concern separation) for good modularity (may be domain dependant)
- A proper combination of tools for each type of system/development stage.
- Def: adaptive/aut. system – when the context changes the system can have an option to move from the preset state to another correct state. Any kind of reactive system is adaptive to some degree, the line between adaptive and simply reactive is not so clear, though the adaptive system has a higher degree of behavioural flexibility.
- Desirable Feature: accept input from a domain expert and incorporate it into the run-time model. Even a longer term goal is to be able to learn from the experience of the system execution itself. (Alexey is not sure about the desirability of this feature anyway!)